

# Sustainable and Efficient Data Collection from WSNs to Cloud

Tian Wang, Yang Li, Guojun Wang, Jiannong Cao, Md Zakirul Alam Bhuiyan, Weijia Jia

**Abstract**—The development of cloud computing pours great vitality into traditional wireless sensor networks (WSNs). The integration of WSNs and cloud computing has received a lot of attention from both academia and industry. However, collecting data from WSNs to cloud is not sustainable. Due to the weak communication ability of WSNs, uploading big sensed data to the cloud within the limited time becomes a bottleneck. Moreover, the limited power of sensor usually results in a short lifetime of WSNs. To solve these problems, we propose to use multiple mobile sinks (MSs) to help with data collection. We formulate a new problem which focuses on collecting data from WSNs to cloud within a limited time and this problem is proved to be NP-hard. To reduce the delivery latency caused by unreasonable task allocation, a time adaptive schedule algorithm (TASA) for data collection via multiple MSs is designed, with several provable properties. In TASA, a non-overlapping and adjustable trajectory is projected for each MS. In addition, a minimum cost spanning tree (MST) based routing method is designed to save the transmission cost. We conduct extensive simulations to evaluate the performance of the proposed algorithm. The results show that the TASA can collect the data from WSNs to Cloud within the limited latency and optimize the energy consumption, which makes the sensor-cloud sustainable.

**Index Terms**—sensor-cloud, mobile sinks, data delivery, energy consumption, sustainability.

## 1 INTRODUCTION

NOWADAYS, the WSNs have been applied in various areas, such as civilian, industry and military (e.g., health monitoring [1], forest fire detection [2], target tracking [3], battlefield surveillance [4], etc.). Sensors in these applications perform diverse functionalities, including data sensing, data acquisition, network security maintenance, and seamlessly monitoring the flow in the network. However, the WSNs are faced with two challenges. On the one hand, the sensors in WSNs are limited in battery and storage ability, which makes it difficult for sensors to be sustainable [5]. On the other hand, the sensors frequently produce intensive data which need to be collected and processed efficiently within a specific time. It is also a challenge for WSNs.

Fortunately, cloud computing technology can be developed to work as a strong backbone for WSNs [6], [7], [8]. With the cloud computing paradigm adopted in WSNs, the performance of WSNs can be improved, such as energy consumption, computing latency, service quality, etc [9]. The birth of sensor-cloud integration is an inevitable trend [10], [11]. In the sensor-cloud integration, users do not need to own sensors. They can simply rent the sensing services. This mechanism significantly reduces the cost of ownership and enables the usage of large scale sensor networks to become affordable. Beyond that, one physical sensor can be projected as multiple services, which improves the efficiency of sensor usage. The nature of sensor-cloud enables

resource sharing. Taking the remote health care application for instance [12], patient health data is collected at remote locations by using the health kits. The health records along with patients' basic information are transmitted to cloud in a timely manner, so that the doctor can give a diagnosis to patient remotely based on health records with the help of cloud. This kind of application improves quality of life for people who live in the poor living conditions.

However, due to the weak transmission ability of WSNs, collecting the sensed data from WSNs to cloud within a limited time becomes a bottleneck [13], [14]. Note that the data collection time is important for some delay-sensitive applications. In the location-based application, if the user's position cannot be uploaded to cloud on time, the location-based service will become failure. For example, in forest fire monitoring application, a large number of dispersed sensors are deployed to continuously monitor the temperature, humidity and gases of forest. As shown in the Figure 1, when the sum of data collection time (CT) and feedback time (FT) is bigger than optimal rescue time, the best rescue time would be missed. The exceeding time (ET) need to be eliminated. Therefore, it is necessary to design a scheme for data collection to improve the throughput of WSNs in sensor-cloud integration.

On the other hand, various mobile elements, such as mobile sensor, mobile agent (MA), mobile sink (MS), have been deployed to collect data in WSNs in recent works [15], [16], [17]. Sensors transmit data to mobile elements directly or through less hops wireless transmission to save energy. For example, in [17], the MA is used to help with forest fire detection. Compared with MA, mobile sink can move to collect data and upload data to cloud directly. Therefore, we use multiple mobile sinks to help with data collection in this paper. For one thing, the MSs have a stronger transmission ability to supplement the communication bot-

- T. Wang is with the Department of Computer Science and Technology, Huaqiao University, Xiamen, Fujian, China; 361021.
- Y. Li is with Huaqiao University.
- G. J. Wang is with Guangzhou University (corresponding author). E-mail: csgjwang@gmail.com
- J. N. Cao is with Hong Kong Polytechnic University.
- M. Z. A. Bhuiyan is with Fordham University.
- W. J. Jia is with Shanghai Jiaotong University.

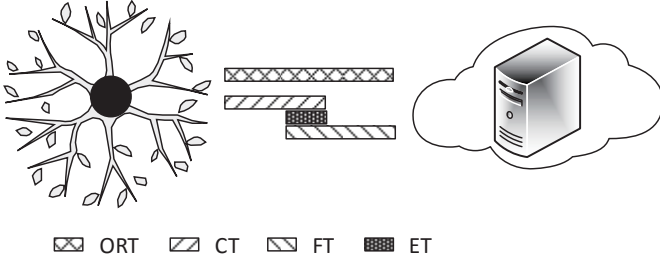


Fig. 1: The influence on forest fire detection brought by delivery delay. When the sum between data collection time (CT) and feedback time (FT) is bigger than optimal rescue time (ORT), the ORT would be missed.

tleneck of WSNs. For another thing, the MSs contribute to balance the forward load of each sensor, which improves the dependability and prolong the lifetime of WSNs. However, due to the limited velocity of mobile sinks, they usually cost lots of time to tour a large sensing field, which extends the data collection time [18].

In this paper, we propose to use MSs helping with data collection in sensor-cloud integration. A time adaptive schedule algorithm TASA is designed to reduce data collection time and make system sustainable. In the TASA, the monitoring area is divided into several sectors equally and each MS is responsible for one sector. In each sector, part of sensors are selected as polling points which construct the trajectories of MSs and all the trajectories are non-overlapping with each other. Besides, the MST is adopted to design transmission route for sensors to save energy consumption. Beyond that, we design two progressive schedule schemes to adjust data collection time and save energy. One focuses on reducing MS's moving time and the other aims at balancing load of each MS. The main contributions of this paper are summarized as follows:

1. Compared with traditional data collection problem in WSNs, the problem of data collection in sensor-cloud integration is more complex. We propose delay constrain problem caused by limited transmission ability of WSNs in sensor-cloud integration.
2. We use multiple MSs to improve the sustainability of sensor-cloud and design a time adaptive algorithm aiming at collecting data from WSNs to Cloud within a specific time, with several provable properties. Beyond that, the energy consumption of sensors is optimized based on property of MST.
3. We conduct extensive simulation experiments to evaluate the performance of the proposed algorithm and the experimental results validate the effectiveness of our proposed algorithm.

The remainder of this paper is organized as follows: Section 2 reviews research related to the work presented herein. Section 3 discusses modeling of our study. The design details for our proposed time adaptive schedule algorithm (TASA) is described in section 4. Section 5 presents the analysis of our proposed algorithm. Section 6 shows simulation results, and the last section concludes our work.

## 2 RELATED WORK

The proliferation of the implementation for low-cost, low-latency, multiple functional sensors has made WSNs a prominent paradigm to monitor the physical world. When the wireless sensor network is deployed, it had been found that there are a mass of data need to be collected. In the traditional WSNs, sensors transmit data to the static sink through multiple hops wireless communication [19], [20]. However, due to vast data transmission work, the sensors that close to the static sink representatively deplete their energy. With the increase of disable nodes, the network connectivity and coverage rate will decrease. In [21], Wu et al. designed energy efficient protocols that sensors consume different energy in different radio states (transmitting, receiving, listening, sleeping, and being idle). They proved that the proposed method considerably reduces energy consumption. In [22], Wang et al. presented an adaptive data gathering scheme by compressive sensing times for wireless sensor networks. They introduced autoregressive (AR) model into the reconstruction of the sensed data. Their extensive test of real data sets and experimental results validate the efficiency and efficacy of the proposed scheme. Although, both two methods reduced energy consumption to some extent, they did not change the model that the sensors which close to sink exhaust energy firstly and lead a short lifetime of WSNs.

With the development of WSNs, the usage of mobile elements are proposed in recent years [23], [24], [25]. Sensors transmit data to mobile elements directly or through less hops wireless transmission, thus sensors can save energy and WSNs achieve a longer lifetime. Two typical mobile elements, mobile agent (MA) and mobile sink, are applied continually. Mobile agent is the device moving to collect sensing data and returning to sink [26], [27], [28]. For example, in [17], the MA is used to help with forest fire detection. The paper introduces a framework equipped with MAs to accelerate detection rate of forest fire with the minimum consumption of energy. Compared with MA, mobile sink can also move to collect sensed data but not need to return to station because it is equipped with the ability of uploading data to cloud, which reduces data collection time. In [29], Tunc C et al. proposed Ring Routing, a novel, distributed, energy efficient mobile sink routing protocol, suitable for time-sensitive applications, which make use of mobile sink to provide load-balanced data delivery and achieve uniform energy consumption across the network. In [30], Arquam M et al. proposed a routing algorithm with sink mobility in hierarchical WSNs to improve network lifetime by eliminating energy holes. They considered upper and lower bound on delay while optimizing sojourn locations and sojourn time. In [31], Hou G et al. designed an efficient path algorithm VG-AFSA based on Virtual Grids to meet most applications' requirements for data latency. They divided nodes into groups with virtual grids to scale down the space of searching optimal set of visited nodes. Their results showed that the algorithm performing well energy effectiveness. However, these researches main focus on the energy efficient problem and ignore the delivery latency. Some researches even sacrifice the delivery time to save energy. On the other hand, these solutions are based

on single mobile sink neglecting the limited throughput of WSNs, which are not applicable for the networks with large number of sensors and stringent time constrain.

In order to address these problems, methods with multiple mobile sinks are designed, which focuses on the issue of multiple mobile sink scheduling to realize limited delay and longer lifetime of large WSNs [32], [33], [34]. For example, in [32], Yi Fan Hu et al. designed an efficient routing recovery protocol with endocrine cooperative particle swarm optimization algorithm (ECP SOA) to establish and optimize the alternative path, which improved the routing protocol robustness and efficiently. Meanwhile, the method reduced the communication overhead and the energy consumption. In [35], Di Francesco M et al. used multiple mobile sinks to assist in data collection. Their results showed that the method can increase the network connectivity and reliability, reduce cost, and decrease energy consumption of individual nodes. Both two papers mainly prolong the network lifetime regardless of the delay constrain. How to balance the network lifetime and transmission delay becomes a key problem. To get the maximum benefit with the minimum cost, it is essential to design a schedule scheme for multiple mobile sinks to collect data in an efficient manner. In [36], Wichmann et al. focused on using faster mobile sinks to reduce the physical collection delay. However, such mobile sinks are often motion-constrained and require smooth path which cannot fits all kinds of application. Moreover, current methods merely consider the data collection from sensors to sink and cannot be applied to the sensor-cloud environment [37]. To the best of our knowledge, we are the first one to consider both delivery delay and energy consumption in sensor-cloud integration [38]. However, our former paper did not deal with the problem that the number of sensor in each sector may be different. In this paper, the algorithm is improved with an additive schedule scheme. No matter the initial number of sensors in each sector, if one of MSs cannot meet the time requirement, some sensors in this sector will change their routes to reduce the load of MS. This method increases the throughput of sensor-cloud integration as well.

### 3 PROBLEM DEFINITIONS AND MODELS

#### 3.1 Problem Definitions

In this paper, we assume the WSN consists of  $N$  sensors, denoted by a set  $S = \{S_1, S_2, \dots, S_N\}$ . The set  $K = \{MS_1, MS_2, \dots, MS_M\}$  represents  $M$  mobile sinks (MSs). For any  $S_i \in S$ , the sensing data rate is  $C$  bytes/s, the single hop latency is  $t$  s and the communication radius is  $R$  m. Any  $MS_i \in K$  can receive data from sensors and upload it to cloud. The throughput from sensors to MS is  $D$  byte/s, and the uploading rate from MS to cloud is  $Q$  byte/s. The velocity of MS is denoted by  $v$  m/s. We focus on the problem of Data Collection from WSNs to Cloud (hereafter referred to as DCWC problem). The goal of this paper aims at reducing the delivery time to a limited time. When the delivery time satisfies the requirement, the energy consumption will be optimized.

A simple example is illustrated in Figure 2, where four mobile sinks are deployed to collect data from WSN to cloud. The circles and small cars stand for fixed sensors and mobile sinks, respectively. The routes of mobile sinks

are shown as dotted line. In Figure 2(a), four routes have the overlapping parts and the deeper sensor's color is, the sensor is visited by more MSs. This phenomenon means that delivery time could be optimized. Then the route of each MS is adjusted in Figure 2(b). As we can see, each MS is responsible for fours sensors. What's more, each sensor is visited only once and the route length of each MS is reduced. Thus, it is essential to design a schedule algorithm to reduce data collection time in sensor-cloud integration.

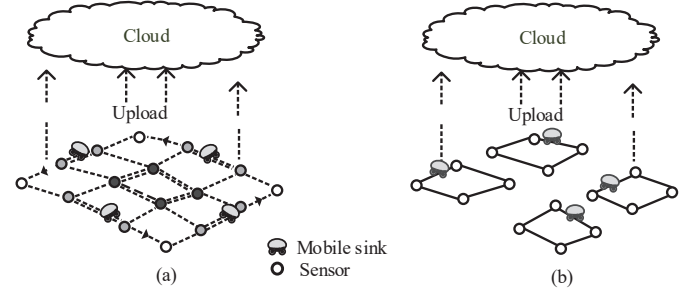


Fig. 2: An example of data collection from WSNs to cloud with multiple mobile sinks. (a) Four mobile sinks are deployed to collect data from WSN to cloud. The route of MSs are random. the sensors' color represents the number of MS visiting this sensor. The deeper the color is, the more MSs visit this sensor. (b) The route of MSs are adjusted and all the sensors are visited by only one MS.

We have the following theorem regarding to the complexity of the DCWC problem:

**Theorem 1.** For the DCWC problem, the problem of designing the optimal algorithm is a NP-hard problem

*Proof of Theorem 1:* We prove this theorem by showing a special case of DCWC, in which  $M=1$ , the sensor transmission radius is zero and uploading time is zero. In this case, the DCWC is equivalent to find a shortest path visiting all of the given sensors. Note that in order to minimize the path length, any optimal solution would not visit the same sensor twice, otherwise we can make it shorter by using triangular inequality. Therefore, finding an optimal solution of this special case of DCWC is equivalent to find an optimal solution of Hamiltonian path problem (i.e., finding a path to visit all sensors with the minimum length), which is a well-known NP-hard problem.  $\square$

#### 3.2 Network Model

We model the connectivity of sensors in WSN as an unoriented-weighted graph  $G = \{V_{se}, E_{se}\}$ , where  $V_{se} = S, E \in \{V_{se} \times V_{se}\}$  is a set of edges, where  $E_{i,j} \in E_{se}$  is the edge if the distance of  $d_{i,j}$  between  $S_i$  and  $S_j$  is smaller than  $R$ . Then the graph  $G$  is transformed to a Minimum Cost Spanning Tree  $MST = \{T_{node}, T_{edge}\}$ , where  $T_{node} = V_{se}$  and  $T_{edge} \subseteq E_{se}$ . Each mobile sink will visit some selected sensors called Polling Points (PPs), denoted by  $\{PP \subseteq S | P_1, P_2, \dots, P_k\}$ . Through visiting all the PPs, the sensory data in WSN would be collected and uploaded to cloud. With multiple mobile sinks synchronously working, the network delivery delay can be estimated as follow:

$$T_{net} = \text{Max} \{T_{MS_1}, T_{MS_2}, \dots, T_{MS_M}\} \quad (1)$$

where  $T_{MS_i}$  is the delivery time of  $MS_i$ , and the network delivery time is the max time among all MSs. As for each individual  $T_{MS_i}$ , it consists of four parts formulated as follows.

$$T_t = \frac{\sum_{j=1}^s C}{D} \quad (2)$$

$$T_d = \frac{\sum_{j=1}^s C}{Q} \quad (3)$$

$$T_h = \sum_{j=1}^s h_j * t \quad (4)$$

$$T_m = \frac{L_{tsp}}{v} \quad (5)$$

$T_t$  is the transmission time from sensor source to sink.  $T_d$  is the uploading time from sink to cloud.  $T_h$  is the multiple-hops delay and  $T_m$  is the traveling time of MS. In all arithmetic expression,  $s$  is the number of sensors assigned to  $MS_i$ . The variable  $h_j$  is the amount of hops from sensor  $S_j$  to  $MS_i$  and  $L_{tsp}$  is the length that  $MS_i$  traveled. We assume each MS can upload data to cloud at any time. Therefore, each  $T_{MS_i}$  can be calculated as formula 6.

$$T_{MS_i} = \frac{\sum_{j=1}^s C}{D} + \sum_{j=1}^s h_j * t + \text{Max} \left( \frac{\sum_{j=1}^s C}{Q}, \frac{L_{tsp}}{v} \right) \quad (6)$$

Based on formula 6, we can conclude that  $h_j$ ,  $L_{tsp}$  and  $s$  are the main external factors affecting  $T_{MS_i}$ , which motivates us to design the algorithm in next section. Note that sensory data can be uploaded when MS is moving according to existed routing algorithm [39].

## 4 TASA: TIME ADAPTIVE SCHEDULE ALGORITHM

### 4.1 Overview of the Algorithm

The design of algorithm can be divided into three sub problems: first, how to distribute task to each MS reasonably; second, how to design the delivery time adaptive mechanism; third, how to reduce the energy consumption when delivery time meets the requirement. In order to address these issues, the solution has corresponding three sub algorithms. In first sub algorithm, called partition and delivery design algorithm (PDDA), the monitoring area is partitioned off  $M$  sectors equally and each MS is responsible for one sector. In each sector, a minimum cost spanning tree is constructed and the edges of tree are the delivery routes of sensors. As for second sub algorithm, called polling point selection algorithm (PPSA), some sensors are selected to serve as polling points (PPs) in each sector. The PPs in the same sector constitute the trajectory of MS. In third algorithm, called time schedule algorithm (TSA), the number of PP is adjusted based on delivery time and the number of sensor in each sector will be adjust to be balance.

Figure 3 shows the main process of TASA. The rectangle area represents the coverage area, and the dotted black circle is the circumscribed circle of rectangle in Figure 3(a). The

area is divided into three sectors equally, denoted by  $MS_1$ ,  $M_2$ , and  $M_3$ . Three MSTs are generated in each sector. The edges of MSTs are delivery paths of sensors. In Figure 3(b), parts of sensors are selected as PPs represented by the gray star nodes. The black dotted lines among stars are the trajectories of MSs. The dotted big arrows means that sensors deliver data to MSs when MSs rest at PPs. When delivery time cannot satisfy the requirement, the trajectory of MS will be adjusted by reducing the number of PP. In Figure 3(c), some sensors no longer serve as PPs and they deliver data to MS via multiple hops, where the dotted small arrows mean data transmission between sensors and sensors. As only one PP left in each sector, MS stays at this PP. All sensors send their data to MS via multiple hops transmission. Considering the different number of sensors in each sector, we design a balanced strategy to adjust the task allocated in each sector.

### 4.2 PDDA: Partition and Delivery Design Algorithm

This sub algorithm is the first part of TASA, which focuses on network partition and delivery route design. In order to simplify partition, the monitoring area is partitioned based on degree. More specifically, we use a min rectangle to include all sensors. However, its difficult to partition the rectangle off  $M$  region with same perimeter. So we partition the coverage area based on its circumscribed circle. Coverage area is equally divided into  $M$  sectors with central angle  $\frac{2\pi}{M}$ . Second, the connectivity among sensors in each sector constitutes  $M$  weighted graphs  $\{G_{MS_i} | i = 1, 2, \dots, M\}$ . Besides, the weight of edge can be calculated by formula 7.

$$E_{i,j} = \sqrt{(S_i.x - S_j.x)^2 + (S_i.y - S_j.y)^2} \quad (7)$$

$S_i.x, S_i.y$  represent the abscissa and ordinate of sensor, respectively. If the value of edge  $E_{i,j}$  is larger than radius  $R$ , the weight of this edge would be set as infinity. Then based on Prim algorithm,  $M$  weighted graphs are transformed to  $M$  minimum cost spanning trees, denoted as  $\{Mst_i | i = 1, 2, \dots, M\}$ . According to the properties of MST, when delivery time  $T_{MS_i}$  is smaller than the latency requirement  $T_{spe}$ , the transmission consumption of sensors is the minimum.

### 4.3 PPSA: Polling Point Selection Algorithm

The polling points make it possible for MS to visit parts of sensors collecting all sensory data. A basic election principle can be described as follow. When  $S_j$  is within the transmission range of  $S_i$ , MS can stay at  $S_i$  to collect both  $S_i$  and  $S_j$ . Consequently, MS can visit part of sensors to complete all data collection.

A reasonable selection of PP can reduce the moving time of MS. In the PDDA, the sensors which construct a MST can be classified into three types: root node, potential PP node and leaf node. Root node is the start point of MS. Potential PP node is kind of sensor which is connected with more than one sensor directly. Leaf node is the sensor which is connected with only one sensor. In Figure 4(a), it is easy known that MS can visit  $S_6, S_2, S_0$  and  $S_4$  to gather all sensory data, then these four sensors are selected as PP like the gray nodes. The dotted line is the trajectory of MS. On

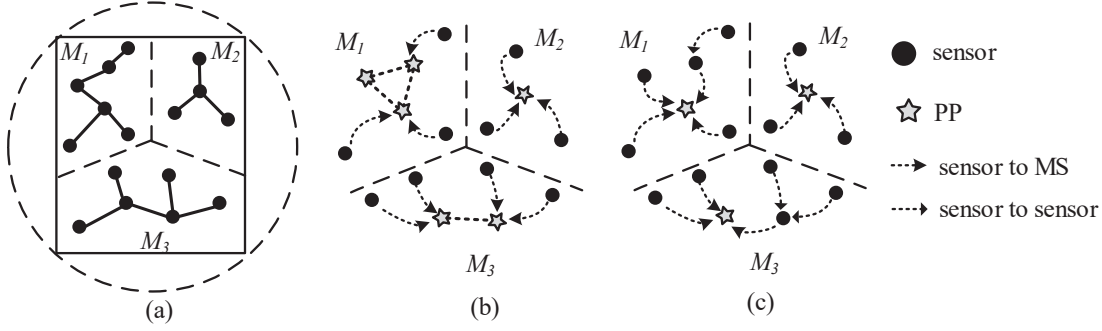


Fig. 3: The basic idea of TASA algorithm. (a) It shows the partition in monitoring area, where the dotted circle is the circumcircle of rectangle and the dotted lines in the rectangle represent the boundary. The boundary is designed based on the degree in circumcircle. (b) Some sensors are selected as PP and all PPs construct the trajectory of MS like the dotted line among stars. (c) Some sensors are no longer PP and the trajectory of MS is redesigned.

---

#### Algorithm 1 PDDA: Partition and Delivery Design Algorithm

---

**Input:** each sensors location  $(S_i.x, S_i.y)$ , the number of sensors is  $N$ . Two sets  $T_{node} = \Phi, T_{edge} = \Phi$ , represent the nodes and the edge of MST, respectively. The  $V_{sec}$  is a set of sensors in same sector;

**Output:**  $M$  graphs  $G(V_G, E_G)$  and MSTs, denoted as  $T(V_T, E_T)$ ;

- 1: **while**  $i, j$  are smaller than  $N$  **do**
  - 2:    $E_{i,j} = \sqrt{(S_i.x - S_j.x)^2 + (S_i.y - S_j.y)^2}$ ;
  - 3:   **if**  $E_{i,j} > R$  **then**
  - 4:      $E_{i,j} = \infty$ ;
  - 5:   **end if**
  - 6: **end while**
  - 7: **while**  $T_{node} \neq V_{sec}$  /\*Prim Algorithm\*/ **do**
  - 8:   find  $MinE_{u,v}, u \in T_{node}, v \in S$ ;
  - 9:   **if**  $v \notin T_{node}$  **then**
  - 10:      $T_{node} = T_{node} \cup v$ ;
  - 11:      $T_{edge} = T_{edge} \cup E_{u,v}$ ;
  - 12:   **end if**
  - 13: **end while**
- 

one hand, the trajectory of MS can be optimized by greedy algorithm. On the other hand, when MS rests at PP, the leaf nodes deliver data to MS through single hop to realize the minimum energy consumption.

Figure 4(b) presents a physical storage structure of MST called children linked list. It shows a clear relationship among sensors in MST.  $S_4$  is the root node.  $S_6, S_0, S_2$  are the potential PP nodes.  $S_1, S_3, S_5, S_7, S_8, S_9$  are the leaf nodes.

#### 4.4 TSA: Time Schedule Algorithm

In before two stages, we have designed an initial trajectory for each MS. It is well-known that the employment of mobile sinks can balance the load of sensors and prolong network lifetime. However, due to limited speed of MS, it always wastes lots of time for moving. To solve this problem, two kinds of schedule schemes are designed to adjust the delivery time until delay requirement are satisfied.

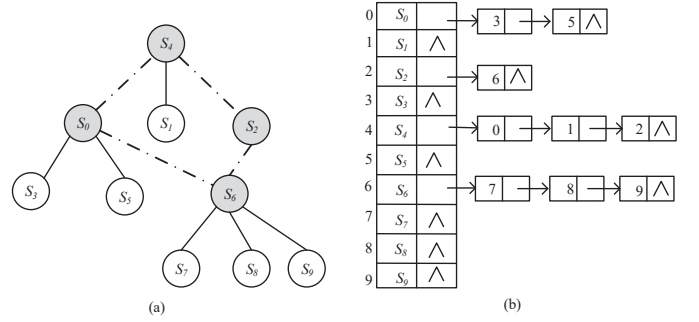


Fig. 4: The strategy of PP selection. (a) In MST, the sensor which connects with more than one other sensor is selected as PP like the gray nodes. (b) This is a physical structure showing a clear relationship among sensors.

**Schedule scheme 1.** In the initial phase, all sensors deliver data to MS via single hop like the white nodes in Figure 5(a). The gray nodes are PPs, and the dotted lines are trajectory of MS. Then we calculate collection time  $T_{MS_i}$  by formula 6. If  $T_{MS_i}$  is greater than  $T_{spet}$ , some PPs will be removed from the set of PP and added into the general sensors set. The PP which is connected with fewer sensors is chosen firstly, such as  $S_4$  in Figure 5(b). Its child node  $S_2$  sends data to MS via  $S_4$  with two hops. Moreover, the trajectory of MS will be redrawn after a round of selection. Until the delivery time meets requirement, the trajectory of MS is fixed. When only one PP left in sector, the moving time of MS is zero, and the delivery time is sum of multiple hops delay and uploading time. As shown in Figure 5(c), the dotted arrows represent the order that PP becomes general sensor.

**Schedule scheme 2:** Different from schedule scheme 1, we consider the factor that the number of sensors is different in each sector. When the latency demand  $T_{tsp}$  is extreme small, parts of sectors may not satisfy the time requirement. Moreover, the network delivery time is the time when the last sensor is collected as shown in formula 1. Consequently, this schedule algorithm aims to balance task of each MS, which can be described in following four steps.

- 1) The sectors are classified into two sets: A and B, where set A includes the sector which delivery time

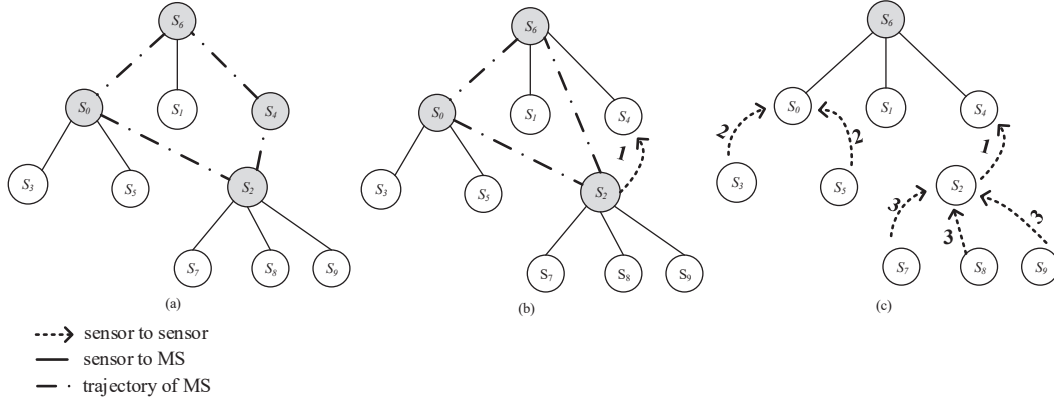


Fig. 5: The schedule process of TSA. (a) It's the initial status of schedule scheme 1. (b) The PP  $S_2$  is transformed to general sensor first. (c) It shows the order that which PP is transformed to be general sensor.

### Algorithm 2 PPSA: Polling Point Selection Algorithm

**Input:** The MST  $T = (T_{node}, T_{edge})$ ; root node:  $u$ ;  $V_i$  represents the set of sensor in sector  $i$ ;  $Vis$  represents the set of sensor which is visited;  
**Output:** the set of PP, denoted by  $\Xi = P_1, P_2, \dots, P_k$  and the visiting order of PPs, denoted by  $\Lambda$ .

- 1:  $Vis = u$ ;
- 2: **while**  $Vis \neq V_i$  **do**
- 3:   **if**  $E_{u,v} \in T_{edge}$  and  $v \neq Vis$  **then**
- 4:      $Vis = \{Vis \cup v\}$ ;
- 5:     **if** sensor  $v$  connects with more than one sensor **then**
- 6:          $\Xi = \Xi \cup v$ ; /\*sensor  $v$  is selected as PP\*/
- 7:     **end if**
- 8:   **end if**
- 9: **end while**
- 10: **while** the set  $\Lambda \neq \Xi$  **do**
- 11:   next  $PP = Min(emph.S_u \mapsto S_w)$ ; /\*find the next minimum path\*/
- 12:    $\Lambda = \{\Lambda \cup S_w\}$ ;
- 13:   updating the start node as  $S_w$ ;
- 14: **end while**

is smaller than  $T_{spe}$ . On the contrary, the set B includes the sector which delivery time is larger than  $T_{spe}$ .

- 2) In set B, the sector with max  $T_{MS_i}$  is selected as Start Sector (SS). In set A, the sector with minimum  $T_{MS_i}$  is selected as End Sector (ES).
- 3) The sensor from SS which is closest to ES delivers its data to MS in ES. In addition, the delivery time of both SS and ES are updated. Then step 1 is repeated.
- 4) Iterate step 2 and step 3 until set A or set B is Null.

## 5 PERFORMANCE ANALYSES

We have proved several properties of the proposed algorithm. The notations that will be used in this part are summarized in Table 1.

**Theorem 2.** In the worst case, the travel distance of each MS satisfies this inequation:  $L_{tsp} \leq \left(1 + \frac{\pi}{M}\right) \times \sqrt{L^2 + H^2}$ .

### Algorithm 3 TSA: time schedule algorithm

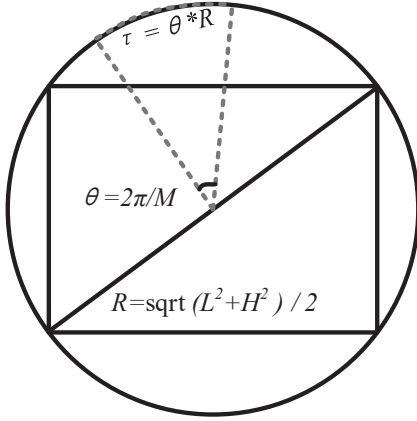
**Input:** The time requirement  $T_{spe}$ ; the delivery time of each sector, denoted by  $\{T_{MS_1}, T_{MS_2}, \dots, T_{MS_M}\}$ ; the set of PP,  $\Xi = \{PP_1, PP_2, \dots, PP_k\}$ ;  
**Output:** the delivery route of each sensor and the trajectory of each MS

- 1: **while**  $i \leq M$  **do**
- 2:   **while**  $T_{MS_i} > T_{spe}$  **do**
- 3:      $PP_i = Min\ degree\ \{PP_1, PP_2, \dots, PP_k\}$ ;
- 4:      $\Xi = \Xi - PP_i$ ; /\*the  $PP_i$  is no longer polling point\*/
- 5:     the trajectory of MS is reprojected;
- 6:   **end while**
- 7: **end while**
- 8: the sectors are divided into two sets, named A and B;
- 9: **if**  $T_{MS_i} > T_{spe}$  **then**
- 10:   put sector  $i$  into set A
- 11: **else** Put sector  $i$  into set B
- 12: **end if**
- 13: **while**  $A \neq \emptyset$  and  $B \neq \emptyset$  **do**
- 14:   Selecting sector  $SS = Max T_{MS_x} \{A\}$  and  $ES = Min T_{MS_y} \{B\}$ ;
- 15:   sensor  $\{S_\xi \in SS | S_\xi \xrightarrow{min} ES\}$  changes its route and deliver data to the MS responsible for ES
- 16:   sets A and B
- 17: **end while**

*Proof of Theorem 2:* Assuming that the sensor coverage area is a rectangle  $L * H$  and its circumscribed circle is shown in Figure 6. The radius of circle is  $R = \sqrt{L^2 + H^2}/2$ . The circle is divided into  $M$  sectors equally based on degree, and the central angle of each sector is  $\theta = 2 * \pi/M$ . The arc of each sector is  $\tau = \theta * R$ . The perimeter of each sector is  $Dist_L = 2 * R + \tau$ . In the worst case, MS have to visit every sensor and the sensors are distributed at margin of sector. Consequently, the max travel distance of MS is the perimeter of sector, namely  $L_{tsp} \leq \left(1 + \frac{\pi}{M}\right) \times \sqrt{L^2 + H^2}$ . This property gives the moving distance of MS an upper bound.  $\square$

TABLE 1: List of notations

Notation	Meaning
M	Number of MSs
n	Number of sensor in WSN
R	The transmission radius of sensor
$L_{tsp}$	The max distance that MS traveled
$\theta$	The central angle of each sector
$\tau$	Arc of each sector
$Dist_L$	Perimeter of each sector
$T_{spe}$	The time requirement
$T_{net}$	The delivery time of network
$\{\mu_1, \mu_2, \dots, \mu_M\}$	$\mu_i$ represents the set of sensors in $\delta_i$
$\{\delta_1, \delta_2, \dots, \delta_M\}$	$\delta_i$ represents the $i$ th sector
$\{\vartheta_1, \vartheta_2, \dots, \vartheta_M\}$	$\vartheta_i$ represents the set of PP in sector $\delta_i$
$\{\gamma_1, \gamma_2, \dots, \gamma_M\}$	$\gamma_i$ represents sensors in sector $\delta_i$
$T_{ideal}$	Theoretical optimum delivery time
$\bar{e}$	Energy consumption for unit length
$edge_{\langle i, j \rangle}$	The weight between $S_i$ and $S_j$
$T_{init}$	The delivery time in PPSA
$E_{init}$	The energy consumption in PPSA
$E_i$	The energy consumption of $\delta_i$
$\ddot{T}$	The actual delivery time
$\ddot{E}$	The actual energy consumption

R: radius  $\theta$ : degree  $\tau$ : arc lengthFig. 6: The coverage area is divided into  $M$  sectors

**Theorem 3.** In the worst case, the time complexity of TASA is  $O(n^3)$ , where  $n$  is the number of sensors.

*Proof of Theorem 3:* In the first sub algorithm PDDA, the coverage area is divided into  $M$  sectors and its time complexity is  $O(1)$ . Then based on Prim algorithm, MST is generated in  $M$  sectors, and its time complexity is  $M \cdot O(n^3)$ . In general case,  $M$  is smaller than  $n$ . Now, we set  $M$  equal to  $n$ , so the time complexity is  $O(n^3)$ . In the second sub algorithm PPSA, the time complexity of electing PP is  $O(n)$ , but time complexity of designing the trajectory of MS based on TSP is  $O(n^3)$  in the worst case. The third sub algorithm TSA consists of two parts. In the first part, assuming that the sensor sets  $\{\mu_1, \mu_2, \dots, \mu_M\}$  all are PPs, then a PP becomes general sensor each round. And the work will be repeated for  $M$  times because of  $M$  sectors, so the complexity of this part can be calculated by  $\sum_1^M \mu_i$  which equals to  $n$ ,

namely,  $O(n)$ . For the second parts, assuming that one sensor changes its transmission route each time, and all the sensor will do this step, so the complexity is  $O(n)$ . Therefore, the time complexity of TASA is  $O(n^3)$  in the worst case.  $\square$

**Theorem 4.** In general value of  $T_{spe}$ , the TASA can realize the adaptive delivery time via schedule scheme 1, and if the time requirement  $T_{ideal}$  almost equals to  $T_{spe}$ , the delivery time can be optimized by schedule scheme 2 that each MS is responsible for same number of sensors. For the time requirement  $T_{spe}$  which is smaller than  $T_{ideal}$ , the delivery time is nearly the minimum in each sector.

*Proof of Theorem 4:* Assuming that sensors in monitoring area is denoted by  $S = \{\mu_1 \cup \mu_2 \cup \dots \cup \mu_M\}$ , where  $\mu_i$  represents a set of sensors in sector  $\delta_i$ .  $\theta_i$  is a set of the polling points in sector  $\delta_i$ , and the  $\gamma_i$  is a set of general sensors in sector  $\delta_i$ . The relationship among  $\mu_i, \theta_i, \gamma_i$  is denoted by  $\mu_i = \theta_i + \gamma_i$ . Based on these notations, the delay model in formula 6 can be simplified further. At the beginning of TSA, all sensors deliver data to MS via single hop, so  $h_i(j = 1, 2, \dots, N)$  equals to 1 and the sum of  $\sum_{j=1}^s C$  equals to  $\gamma \cdot C$ .  $L_{tsp}$  is the travel distance of visiting all PPs in  $\theta_i$ , and we assume that MS can upload data to cloud when MS is moving. Therefore, the delivery time of MS can be simplified as equation 8.

$$T_{MS_i} = \frac{\gamma_i \cdot C}{D} + \gamma_i \cdot t + \text{Max}\left(\frac{L_{tsp}}{V}, \frac{\gamma_i \cdot C}{Q}\right) \quad (8)$$

As the delivery time cant meet application requirement, the TSA would adjust the path of MS to reduce moving time. For the first case, the time constraints  $T_{spe}$  is smaller than  $T_{ideal}$ . Until the  $T_{MS_i}$  is smaller than  $T_{spe}$ , parts of PPs are transformed to general sensors and both two sets  $\gamma_i$  and  $\theta_i$  are not null. In this situation,  $T_{MS_i}$  is smaller than  $T_{spe}$  and  $h_j$  equals to 1 always. Furthermore, when  $T_{spe}$  nearly equals to  $T_{ideal}$ , the goal of problem can be regarded as minimizing the delivery time. Therefore, the PP would be fixed in each sector and most of sensors deliver data to MS via multiple hops such as Figure 5c, where  $\theta_i = 1, \mu_i = \gamma_i + 1$ . One thing to note is that  $T_{tsp}$  equal to zero, then delivery time can be showed like equation 9.

$$T_{MS_i} = \frac{\gamma_i \cdot C}{D} + \frac{\gamma_i \cdot C}{Q} + \sum_{j=1}^{\gamma_i} h_j \cdot t \quad (9)$$

According to the analysis of formula 9, it is obvious that  $\gamma_i$  is the key factor influencing delivery rime. Through the schedule scheme 2 in TSA, the final status is that all the MSs are responsible for same number of sensors, denoted as  $\gamma_1 = \gamma_2 = \dots = \gamma_M$ . Therefore, if  $T_{spe}$  nearly equals to  $T_{ideal}$  or  $T_{spe}$  is much smaller than  $T_{ideal}$ , the theorem 4 is validated, denoted as  $T_{MS_i} \approx T_{ideal}$ .  $\square$

**Theorem 5.** If the solution for designing the trajectory of MS is the optimal in PPSA, the sensor energy consumption is the minimum.

*Proof of Theorem 5:* The initial delivery time and network energy consumption can be calculated by equation 10-11.

$$T_{init} = \frac{\sum_{j=1}^s C}{D} + \sum_{j=1}^s h_j * t + \text{Max}\left(\frac{\sum_{j=1}^s C}{Q}, \frac{L_{tsp}}{v}\right) \quad (10)$$

$$E_{init} = \bar{e} * \sum_{i=1}^M \sum T_{edge} \quad (11)$$

Based on the property of MST, the sum length of edges is the minimum in each sector, namely  $\sum T_{edge}$  is minimum. Therefore, at the initial stage, the energy consumption  $E_{init}$  is the minimum in each sector. Now giving a PP set  $\theta_k = \{PP_1, PP_2, \dots, PP_i, \dots, PP_r\}$ , the child nodes of  $PP_i$  is denoted as  $\{S_a, S_b, \dots, S_k\}$ .  $\Delta t$  is the reduced time and  $\Delta E$  is the increased energy consumption via TSA. The travel distance of MS will decrease, denoted by  $\Delta L = L_{tsp} - \sum_{j=i-1}^{i+1} d_{i,j}$  ( $j \neq i$ ), where  $d_{i,j}$  is the distance between  $PP_i$  and  $PP_j$ . The hop will increase, denoted as  $\Delta h = \sum h_w$  ( $w = S_a, S_b, \dots, S_k$ ). The transmission distance of sensors will increase, denoted as  $\bar{L} = \sum_{i=1}^{\Delta h} l_i$ , where the notation  $l_i$  is the length of each hop. Consequently, the decreased delivery time can be denoted as  $\Delta T = \frac{\Delta L}{v} - \Delta h * t$ . The increased energy consumption can be denoted as  $\Delta E = \bar{e} * \sum_{i=1}^{\Delta h} l_i * t$ . The relationship between actual energy consumption  $\bar{E}$  and actual delivery time  $\bar{T}$  satisfy the equations 12-13.

$$\bar{T} = T_{init} - \frac{L_{tsp} - \sum_{j=i-1}^{i+1} d_{i,j}}{v} + \sum h_w * t, w = S_a, S_b, \dots, S_k \quad (12)$$

$$\bar{E} = E_{init} + \bar{e} * \sum_{i=1}^{\Delta h} l_i * t, w = S_a, S_b, \dots, S_k \quad (13)$$

Based on theorem 4, the delivery time  $\bar{T}$  can be optimized. When solution for designing trajectory of MS in PPSA is optimal, the value of  $\sum h_w$  is optimal. According to the property of MST, the value  $\sum L_i$  is the minimum and the value of  $\Delta E$  is the minimum in each sector. Therefore, the value of  $\bar{E}$  is optimal with the minimum value of  $E_{init}$ .  $\square$

## 6 EVALUATION

### 6.1 Experimental Environments

To validate the effectiveness of our proposed algorithm, we conducted extensive simulations using MATLAB 2015a. We built a wireless sensor network consisting of 100 sensors deployed in a  $100m \times 100m$  rectangle region. The data generating rate of each sensor is 5 byte per second. The transmission range of each sensor is 3 meters and its initial energy capacity is 30 J. The energy consuming rate for the transmitting is  $6 \times 10^{-7} (J/bit)$  and for receiving rate is  $3 \times 10^{-7} (J/bit)$ . The speed of the mobile sink is  $3(m/s)$ . The data uploading rate from the mobile sink to Cloud is 50 (byte/s). In the general case, there are 5 mobile sinks deployed in the WSNs.

As part of the evaluation, two existing algorithms are also considered for comparison. The first algorithm is

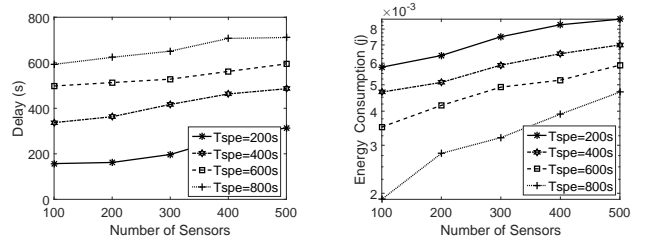


Fig. 7: (a) Number of Sensor vs. Delay (b) Number of Sensor vs. Energy Consumption

EMMS [40]. In this algorithm, multiple sinks were controlled to visit all the sensors and the tour of each mobile sink is closed. The second is SG-MIP [41]. This algorithm iteratively partitioned the monitoring area and selected near-optimal rout for mobile sinks. In the simulations, these three algorithms are compared based on four metrics, i.e., the delivery delay, the energy consumption, the distances that mobile sink traveled and the lifetime of WSNs. The delivery delay is the max delivery time among all mobile sinks. The energy consumption represents the max energy consumption among all sensors. The energy consumption of each sensor is calculated based on the energy model in [42].

### 6.2 Experimental Results

Figure 7 demonstrates the delivery delay and energy consumption under the scenarios with different number of sensors. As shown in Figure 7 (a), when the number of sensor increases from 100 to 500, the delivery delay shows a rising trend. Due to the stringent deadline, when  $T_{spe}$  is smaller than 400 s, the delay requirement cannot be guaranteed. As the  $T_{spe}$  increases, our algorithm performs well with different number of sensors, which is consistent well with the theoretical analysis (Theorem 4). Figure 7 (b) shows the energy consumption for data collection. No matter the value of latency requirement is, as the number of sensors increases, the energy consumption decreases gradually since more sensors can deliver data to MS through the less hops. Moreover, the lower value of latency requirement is, the higher energy consumption is. This is because the number of PPs will be cut down to shorten the travel time of MS, which means more sensors will deliver data via multiple hops.

Figure 8 shows the delivery delay and energy consumption under the scenarios with different number of mobile sinks. In the Figure 8 (a), when the number of mobile sinks increases, the delay with different demand all decrease. It is worth noting that the delay value have a nearly linear variation as the number of MSs is from 10 to 25 and  $T_{spe}$  is 800s. This phenomenon reflects the step in PPSA. When all the sensors send data to MS through single hop and the delivery time is smaller than  $T_{spe}$ , the collection time from sensors to mobile sinks is constant. And the difference is that more MSs uploading data to cloud. As shown in Figure 8 (b), the energy consumption decreases with the MSs increased. This is because the more MSs are deployed to balance the load of each sensor. Besides, more sensors send data to MSs directly.



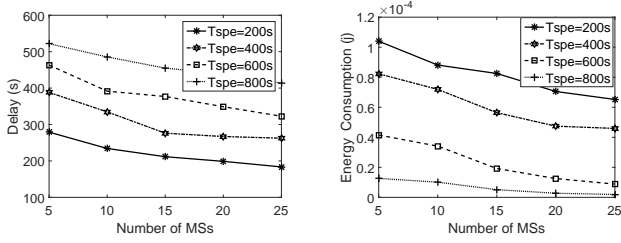


Fig. 8: (a) Number of MSs vs. Delay (b) Number of MSs vs. Energy Consumption

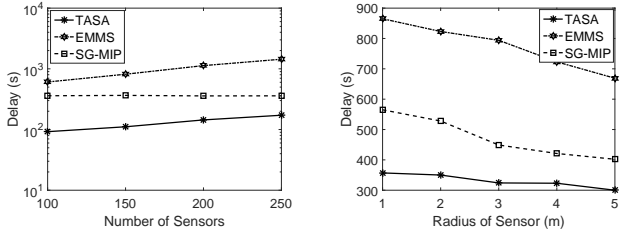


Fig. 9: (a) Number of sensor vs. Delay (b) Number of MSs vs. Delay

We now evaluate the algorithms on delay. As shown in Figure 9 (a), with the increase of the number of sensors, our proposed algorithm TASA achieves the best performance among these three algorithms. Specifically, the delay achieved by SG-MIP is about twice that of our algorithm. When the number of sensors is 250, the delay generated by EMMS is extreme high. In Figure 9 (b), when the number of MSs increases, all algorithms perform better, but TASA performs the best.

After comparing the effect of TASA with another two algorithms, EMMS and SG-MIP, we find two reasons for why our algorithm performs better. First, the radius of sensor is a key element affecting the performance of algorithm. As shown in Figure 10 (a), it is evident that the delay decreases with a bigger radius of sensor for all algorithms. Obviously, the influence of radius on EMMS is the most apparent. And it testifies that the TASA performs a better compatibility. For further clarity, we present the distances MSs traveled in all algorithms in Figure 10 (b). It is seen that TASA has the shortest distance and smallest difference among these three algorithms. This is because TASA can adjust the trajectory to fit the latency requirement.

We now evaluate the effectiveness of algorithm on en-

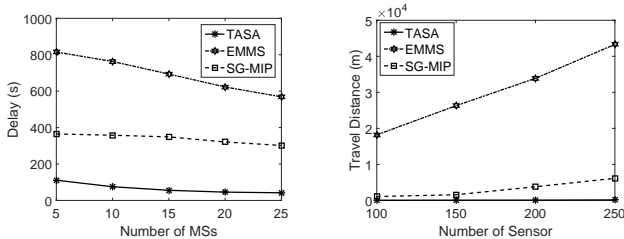


Fig. 10: (a) Radius of sensor vs. Delay (b) Number of sensor vs. Travel Distance of MS

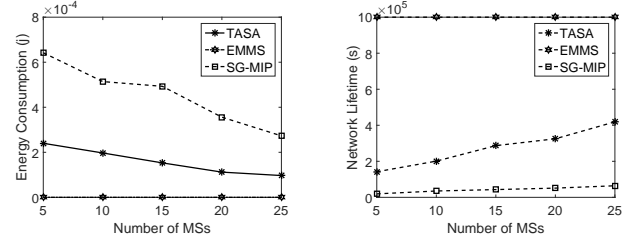


Fig. 11: (a) Number of MS vs. Energy Consumption (b) Number of MS vs. Network Lifetime

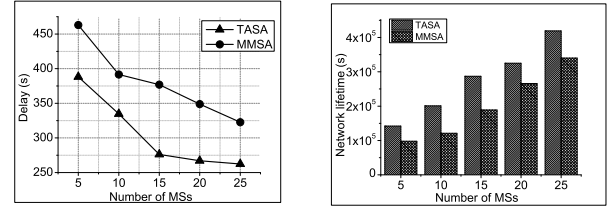


Fig. 12: (a) Number of MSs vs. Delay (b) Number of Sensor vs. Network Lifetime

ergy consumption and network lifetime. In Figure 11 (a), the energy consumption achieved by EMMS stays steady due to single hop transmission of all the sensors. Beyond that, the energy consumption in TASA and SG-MIP decrease when the number of MS is added, and TASA realize the lower energy consumption. Figure 11 (b) shows the trend of network lifetime with the variance of number of MSs. As more MSs are employed in WSNs, the lifetime becomes longer. This is because sensors have more opportunities to communicate with MS directly. In contrast, the lifetime achieved by SG-MIP is shorter than our algorithm.

In the Figure 12, we present the comparison result between TASA and MMSA which was proposed in [38]. Figure 12 (a) presents the comparison on delay with the increased number of mobile sink. Figure 12 (b) implies the tendency of network lifetime with the increased number of mobile sink. Both two results obviously show that the improved algorithm TASA performs better than MMSA.

## 7 CONCLUSION

The development of cloud computing brings new technologies to traditional wireless sensor network, such as high-speed computing power, big data storage and remote service, which makes some applications become possible in our life. It is an inexorable trend to integrate WSNs with cloud computing. With the cloud computing paradigm adopted in WSNs, the performance of WSNs can be improved, such as energy consumption, computing latency, service quality, etc. However, due to the weak communication ability of WSNs, how to upload the mass sensed data to the cloud within the limited time becomes a bottleneck in sensor cloud integration system. Most traditional methods of mobile data collection are delay tolerant, which is not appropriate for data collection in sensor-cloud integration. In this paper, we have studied the data collection problem from WSNs to cloud with multiple mobile sinks and formulated it

as a constrained optimization problem, which is proved to be NP-hard. We designed TASA algorithm which is a polynomial time algorithm and with provable performance. The performance of the proposed method is also validated through simulations. Simulation results demonstrate that the proposed algorithm can adjust the delivery delay, reduce energy consumption significantly and improve the system sustainability, which contributes to the integration of WSNs and Cloud.

## ACKNOWLEDGMENTS

Above work was supported in part by grants from Chinese National Research Fund (NSFC) Key Project No. 61532013 and National China 973 Project No. 2015CB352401 and Shanghai Scientific Innovation Act of STCSM No.15JC1402400 and 985 Project of Shanghai Jiao Tong University with No. WF220103001 and the National Natural Science Foundation (NSF) of China under Grant Nos. 61672441 and 61632009 and Operation Platform and the Foster Project for Graduate Student in Research and Innovation of Huaqiao University (No. 1511414005).

## REFERENCES

- [1] S. Chandrasekaran and T. Chitambaram, "Health monitoring of offshore structures using wireless sensor network: experimental investigations," in *SPIE Smart Structures and Materials+ Nondestructive Evaluation and Health Monitoring*. International Society for Optics and Photonics, 2016, pp. 980 416–980 416.
- [2] D. C. Harrison, W. K. Seah, and R. Rayudu, "Rare event detection and propagation in wireless sensor networks," *ACM Computing Surveys (CSUR)*, vol. 48, no. 4, p. 58, 2016.
- [3] T. Wang, Z. Peng, J. Liang, S. Wen, M. Z. A. Bhuiyan, Y. Cai, and J. Cao, "Following targets for mobile tracking in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 12, no. 4, p. 31, 2016.
- [4] F. T. Jaigirdar and M. M. Islam, "A new cost-effective approach for battlefield surveillance in wireless sensor networks," in *2016 International Conference on Networking Systems and Security (NSysS)*. IEEE, 2016, pp. 1–6.
- [5] Y. Zhang, X. Sun, and B. Wang, "Efficient algorithm for k-barrier coverage based on integer linear programming," *China Communications*, vol. 13, no. 7, pp. 16–23, 2016.
- [6] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2016.
- [7] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2706–2716, 2016.
- [8] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," 2015.
- [9] Y.-J. Ren, J. Shen, J. Wang, J. Han, and S.-Y. Lee, "Mutual verifiable provable data auditing in public cloud storage," *Journal of Internet Technology*, vol. 16, no. 2, pp. 317–323, 2015.
- [10] A. Gupta and N. Mukherjee, "Implementation of virtual sensors for building a sensor-cloud environment," in *2016 8th International Conference on Communication Systems and Networks (COMSNETS)*. IEEE, 2016, pp. 1–8.
- [11] M. J. Guezguez, S. Rekhis, and N. Boudriga, "A sensor cloud architecture for healthcare applications," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. ACM, 2016, pp. 612–617.
- [12] S. Saha, R. Das, S. Datta, and S. Neogy, "A cloud security framework for a data centric wsn application," in *Proceedings of the 17th International Conference on Distributed Computing and Networking*. ACM, 2016, p. 39.
- [13] J. Zhang, J. Long, G. Zhao, and H. Zhang, "Minimized delay with reliability guaranteed by using variable width tiered structure routing in wsn," *International Journal of Distributed Sensor Networks*, vol. 2015, p. 4, 2015.
- [14] D. Kim, R. Uma, B. H. Abay, W. Wu, W. Wang, and A. O. Tokuta, "Minimum latency multiple data mule trajectory planning in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 4, pp. 838–851, 2014.
- [15] M. Zhao, Y. Yang, and C. Wang, "Mobile data gathering with load balanced clustering and dual data uploading in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 770–785, 2015.
- [16] T. Wang, Z. Peng, C. Wang, Y. Cai, Y. Chen, H. Tian, J. Liang, and B. Zhong, "Extracting target detection knowledge based on spatiotemporal information in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2016, p. 1, 2016.
- [17] K. Trivedi and A. K. Srivastava, "An energy efficient framework for detection and monitoring of forest fire using mobile agent in wireless sensor networks," in *IEEE International Conference on Computational Intelligence and Computing Research*, 2015, pp. 1–4.
- [18] T. Wang, Z. Peng, Y. Chen, Y. Cai, and H. Tian, "Continuous tracking for mobile targets with mobility nodes in wsn," in *Smart Computing (SMARTCOMP), 2014 International Conference on*. IEEE, 2014, pp. 261–268.
- [19] J. Shen, H.-W. Tan, J. Wang, J.-W. Wang, and S.-Y. Lee, "A novel routing protocol providing good transmission reliability in underwater sensor networks," *Journal of Internet Technology*, vol. 16, no. 1, pp. 171–178, 2015.
- [20] D. V. Jose and G. Sadashivappa, "A novel scheme for energy enhancement in wireless sensor networks," in *Computation of Power, Energy Information and Communication (ICCPEIC), 2015 International Conference on*. IEEE, 2015, pp. 0104–0109.
- [21] Y. Wu, X. Y. Li, Y. H. Liu, and W. Lou, "Energy-efficient wake-up scheduling for data collection and aggregation," *Parallel & Distributed Systems IEEE Transactions on*, vol. 21, no. 2, pp. 275–287, 2010.
- [22] J. Wang, S. Tang, B. Yin, and X.-Y. Li, "Data gathering in wireless sensor networks through intelligent compressive sensing," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 603–611.
- [23] R. Zhang, J. Pan, D. Xie, and F. Wang, "Ndcmc: A hybrid data collection approach for large-scale wsn using mobile element and hierarchical clustering," 2015.
- [24] F. Restuccia, G. Anastasi, M. Conti, and S. K. Das, "Analysis and optimization of a protocol for mobile element discovery in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 9, pp. 1942–1954, 2014.
- [25] J. S. Prashanth and S. V. Nandury, "Cluster-based rendezvous points selection for reducing tour length of mobile element in wsn," in *Advance Computing Conference (IACC), 2015 IEEE International*. IEEE, 2015, pp. 1230–1235.
- [26] J. J. Chen, C. F. Xing, Z. F. Cheng, and X. B. Wang, "Node information collection routing algorithm using mobile agent in wsn initialization stage," *Applied Mechanics and Materials*, vol. 462–463, pp. 64–67, 2013.
- [27] T. Suetsugu, S. Matsunaga, T. Torikai, and H. Furukawa, "Effective data collection scheme by mobile agent over wireless sensor network," *Ieice Transactions on Communications*, vol. 2015, no. 3, pp. 1–6, 2015.
- [28] M. Dong, K. Ota, L. T. Yang, S. Chang, H. Zhu, and Z. Zhou, "Mobile agent-based energy-aware and user-centric data collection in wireless sensor networks," *Computer Networks*, vol. 74, no. PB, pp. 58–70, 2014.
- [29] C. Tunca, S. Isik, M. Y. Donmez, and C. Ersoy, "Distributed mobile sink routing for wireless sensor networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 877–897, 2014.
- [30] M. Arquam, C. Gupta, and M. Amjad, "Delay constrained routing algorithm for wsn with mobile sink," in *Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on*. IEEE, 2014, pp. 1449–1454.
- [31] G. Hou, X. Wu, C. Huang, and Z. Xu, "A new efficient path design algorithm for wireless sensor networks with a mobile sink," in *The 27th Chinese Control and Decision Conference (2015 CCDC)*. IEEE, 2015, pp. 5972–5977.
- [32] Y.-F. Hu, Y.-S. Ding, L.-H. Ren, K.-R. Hao, and H. Han, "An endocrine cooperative particle swarm optimization algorithm for

- routing recovery problem of wireless sensor networks with multiple mobile sinks," *Information Sciences*, vol. 300, pp. 100–113, 2015.
- [33] P. Madhumathy and D. Sivakumar, "Enabling energy efficient sensory data collection using multiple mobile sink," *China Communications*, vol. 11, no. 10, pp. 29–37, 2014.
- [34] S. Xie and Y. Wang, "Construction of tree network with limited delivery latency in homogeneous wireless sensor networks," *Wireless personal communications*, vol. 78, no. 1, pp. 231–246, 2014.
- [35] M. Di Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: A survey," *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, no. 1, p. 7, 2011.
- [36] A. Wichmann and T. Korkmaz, "Smooth path construction and adjustment for multiple mobile sinks in wireless sensor networks," *Computer Communications*, vol. 72, pp. 93–106, 2015.
- [37] F. Zhangjie, S. Xingming, L. Qi, Z. Lu, and S. Jiangang, "Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Transactions on Communications*, vol. 98, no. 1, pp. 190–200, 2015.
- [38] Y. Li, T. Wang, G. Wang, J. Liang, and H. Chen, "Efficient data collection in sensor-cloud system with multiple mobile sinks," in *Advances in Services Computing: 10th Asia-Pacific Services Computing Conference, APSCC 2016, Zhangjiajie, China, November 16-18, 2016, Proceedings*. Springer, 2016, pp. 130–143.
- [39] T. Wang, W. Jia, B. Zhong, H. Tian, and G. Zhang, "Bluecat: An infrastructure-free system for relative mobile localization," *Adhoc & Sensor Wireless Networks*, 2015.
- [40] J. Wang, Y. Zhang, Z. Cheng, and X. Zhu, "Emip: energy-efficient itinerary planning for multiple mobile agents in wireless sensor network," *Telecommunication Systems*, vol. 62, no. 1, pp. 93–100, 2016.
- [41] J. Shi, X. Wei, and W. Zhu, "An efficient algorithm for energy management in wireless sensor networks via employing multiple mobile sinks," *International Journal of Distributed Sensor Networks*, vol. 2016, p. 9, 2016.
- [42] C. Zhu, V. C. Leung, L. T. Yang, and L. Shu, "Collaborative location-based sleep scheduling for wireless sensor networks integrated with mobile cloud computing," *IEEE Transactions on Computers*, vol. 64, no. 7, pp. 1844–1856, 2015.



**Tian Wang** received his BSc and MSc degrees in Computer Science from the Central South University in 2004 and 2007, respectively. He received his PhD degree in City University of Hong Kong in 2011. Currently, he is an associate professor in the National Huaqiao University of China. His research interests include wireless networks, Internet of things and mobile computing.



**Yang Li** received her BSc degree in Computer Science from China Three Gorges University in 2015. Currently, she is a graduate student in the National Huaqiao University. Her research interests include wireless networks, Internet of things and mobile computing.



**Guojun Wang** received B.Sc. in Geophysics, M.Sc. in Computer Science, and Ph.D. in Computer Science, from Central South University, China. He is Chair and Professor of Department of Computer Science at Central South University. He is Director of Trusted Computing Institute at Central South University. He has been an Adjunct Professor at Temple University, USA; a Visiting Scholar at Florida Atlantic University, USA; a Visiting Researcher at the University of Aizu, Japan; and a Research Fellow at the Hong Kong Polytechnic University. His research interests include network and information security, Internet of things, and cloud computing. He is a senior member of CCF, and a member of IEEE, ACM, and IEICE.



**Jiannong Cao** Jiannong Cao received the BSc degree in computer science from Nanjing University, China, and the MSc and PhD degrees in computer science from Washington State University, Pullman. He is currently a professor and associate head in the Department of Computing at Hong Kong Polytechnic University. His research interests include mobile and pervasive computing, computer networking, parallel and distributed computing, and fault tolerance. His recent research has focused on wireless networks and mobile and pervasive computing systems, developing test-beds, protocols, middleware, and applications. He has served as an associate editor and a member of editorial boards of several international journals, including *Pervasive and Mobile Computing*, *Wireless Communications and Mobile Computing*, *Peer-to-Peer Networking and Applications*, and the *Journal of Computer Science and Technology*. He has also served as a chair or member of organizing/program committees for many international conferences, including PERCOM, ICDCS, IPDPS, ICPP, RTSS, SRDS, MASS, PRDC, ICC, GLOBECOM, and WCNC. He is a senior member of China Computer Federation, the IEEE, the IEEE Computer Society, and the IEEE Communication Society, and a member of ACM.



**Md Zakirul Alam Bhuiyan** received the PhD degree and the M. Eng. degree from Central South University, China, in 2009 and 2013 respectively, and the BSc degree from International Islamic University Chittagong, Bangladesh, in 2005, all in Computer Science and Technology. He is currently an assistant professor (research) in the Department of Computer and Information Sciences at Temple University. He is a member of the Center for Networked Computing (CNC). Earlier, he worked as a post-doctoral fellow at the Central South University, China, a research assistant at the Hong Kong PolyU, and a software engineer in industries. His research focuses on dependable cyber physical systems, wireless sensor network applications, network security, and sensor-cloud computing. He has served as a managing guest editor, program chair, workshop chair, publicity chair, TPC member, and reviewer of international journals/conferences. He is a member of IEEE and a member of ACM.



**Weijia Jia** received the BSc and MSc degrees from Central South University, Changsha, China and the M Applied Sci and PhD degrees from the Polytechnic Faculty of Mons, Belgium, all in computer science. His research interests include wireless communication and networks, distributed systems, multicast, and (pioneer) anycast QoS routing protocols for the Internet. In these fields, he has more than 300 publications in international journals, books/chapters, and refereed international conference proceedings. He is the chair professor of the Central South University and a guest professor at the University of Science and Technology of China, Beijing Jiaotong University, and Jinan University, Guangzhou, China. He has served as an editor and guest editor for international journals and as a PC chair and PC member/keynote speaker for various IEEE international conferences. He has been listed in Marquis Who's Who (VIP) in the World (2000-2008). He is a senior member of the IEEE and a member of the ACM